15-441 15-641 Computer Networking

Lecture 8 – Internet Protocol, NATs and Tunnels

Peter Steenkiste

Fall 2014

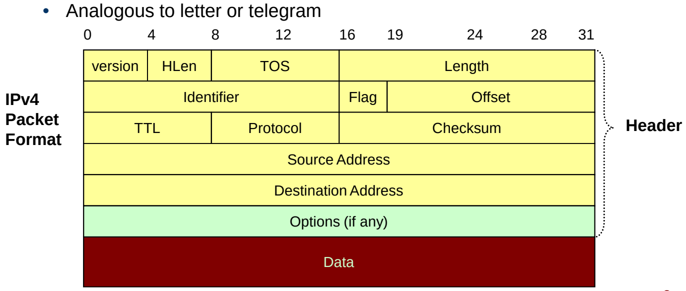
www.cs.cmu.edu/~prs/15-441-F14

Outline
The Good, the Bad and the Ugly

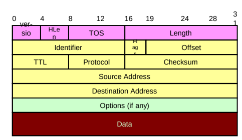
- IP protocol
- NATs
- Tunnels

IP Service Model

- Low-level communication model provided by Internet
- Datagram
  - Each packet self-contained
    - All information needed to get to destination
    - No advance setup or connection maintenance
  - Analogous to letter or telegram



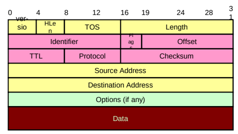
IPv4 Header Fields



- Version: IP Version
  - 4 for IPv4
- HLen: Header Length
  - 32-bit words (typically 5)
- TOS: Type of Service
  - Priority information
- Length: Packet Length
  - Bytes (including header)
- Header format can change with versions
  - First byte identifies version
- Length field limits packets to 65,535 bytes
  - In practice, break into much smaller packets for network performance considerations

IPv4 Header Fields

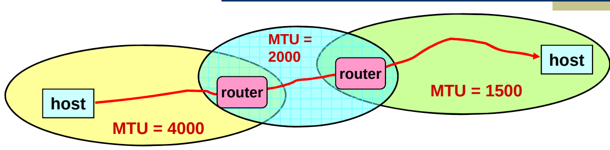
- Identifier, flags, fragment offset → used for fragmentation
- Time to live
  - Must be decremented at each router
  - Packets with TTL=0 are thrown away
  - Ensure packets exit the network
- Protocol
  - Demultiplexing to higher layer protocols
  - TCP = 6, ICMP = 1, UDP = 17…
- Header checksum
  - Ensures some degree of header integrity
  - Relatively weak – 16 bit
- Source and destination IP addresses
- Options
  - E.g. Source routing, record route, etc.
  - Performance issues
    - Poorly supported



IP Delivery Model

- *Best effort service*
  - Network will do its best to get packet to destination
- Does NOT guarantee:
  - Any maximum latency or even ultimate success
  - Sender will be informed if packet doesn't make it
  - Packets will arrive in same order sent
  - Just one copy of packet will arrive
- Implications
  - Scales very well (really, it does)
  - Higher level protocols must make up for shortcomings
    - Reliably delivering ordered sequence of bytes → TCP
  - Some services not feasible (or hard)
    - Latency or bandwidth guarantees

IP Fragmentation

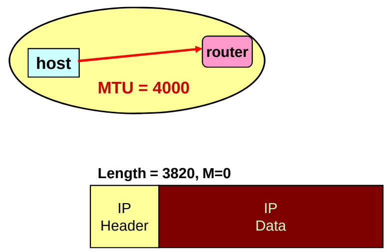


- Every network has own Maximum Transmission Unit (MTU)
  - Largest IP datagram it can carry within its own packet frame
    - E.g., Ethernet is 1500 bytes
  - Don't know MTUs of all intermediate networks in advance
- IP Solution
  - When hit network with small MTU, router fragments packet
  - Destination host reassembles the paper – why?

Fragmentation Related Fields

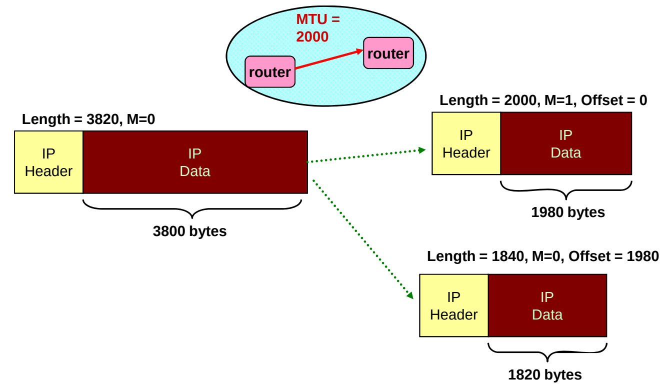
- Length
  - Length of IP fragment
- Identification
  - To match up with other fragments
- Flags
  - Don't fragment flag
  - More fragments flag
- Fragment offset
  - Where this fragment lies in entire IP datagram
  - Measured in 8 octet units (13 bit field)

IP Fragmentation Example #1



9

IP Fragmentation Example #2



10

Fragmentation is Harmful

- Uses resources poorly
  - Forwarding costs per packet
  - Best if we can send large chunks of data
  - Worst case: packet just bigger than MTU
- Poor end-to-end performance
  - Loss of a fragment
- Path MTU discovery protocol → determines minimum MTU along route
  - Uses ICMP error messages
- Common theme in system design
  - Assure correctness by implementing complete protocol
  - Optimize common cases to avoid full complexity

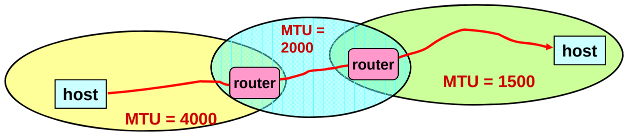
11

Internet Control Message Protocol (ICMP)

- Short messages used to send error & other control information
- Examples
  - Ping request / response
    - Can use to check whether remote host reachable
  - Destination unreachable
    - Indicates how packet got & why couldn't go further
  - Flow control
    - Slow down packet delivery rate
  - Redirect
    - Suggest alternate routing path for future messages
  - Router solicitation / advertisement
    - Helps newly connected host discover local router
  - Timeout
    - Packet exceeded maximum hop limit

12

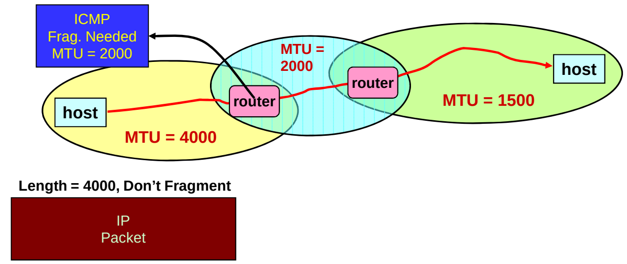
IP MTU Discovery with ICMP



- Typically send series of packets from one host to another
- Typically, all will follow same route
  - Routes remain stable for minutes at a time
- Makes sense to determine path MTU before sending real packets
- Operation: Send max-sized packet with “do not fragment” flag set
  - If encounters problem, ICMP message will be returned
    - “Destination unreachable: Fragmentation needed”
    - Usually indicates MTU problem encountered
- ICMP abuse? Other solutions?

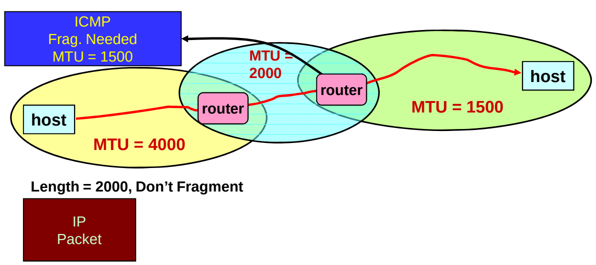
13

IP MTU Discovery with ICMP



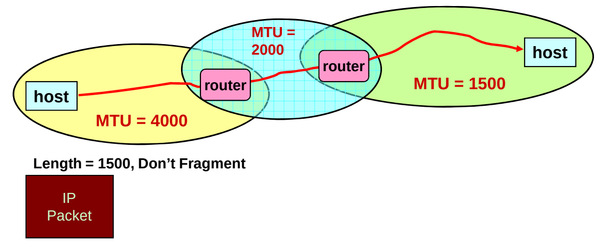
14

IP MTU Discovery with ICMP



15

IP MTU Discovery with ICMP



- When successful, no reply at IP level
  - “No news is good news”
- Higher level protocol might have some form of acknowledgement

16

Important Concepts

- Base-level protocol (IP) provides minimal service level
  - Allows highly decentralized implementation
  - Each step involves determining next hop
  - Most of the work at the endpoints
- ICMP provides low-level error reporting

- IP forwarding → global addressing, alternatives, lookup tables
- IP addressing → hierarchical, CIDR
- IP service → best effort, simplicity of routers
- IP packets → header fields, fragmentation, ICMP

17

Outline

- IP protocol
- NATs
- Tunnels

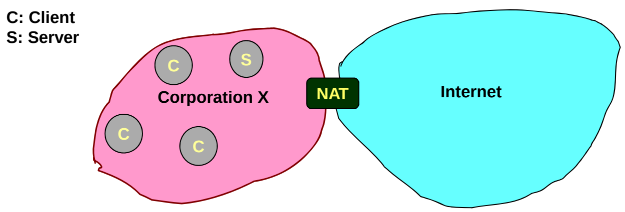
18

Altering the Addressing Model

- Original IP Model: Every host has unique IP address
- Implications
  - Any host can communicate with any other host
  - Any host can act as a server
    - Just need to know host ID and port number
- System is open – complicates security
  - Any host can attack any other host
  - Possible to forge packets
    - Use invalid source address
- Places pressure on the address space
  - Every host requires "public" IP address

19

Challenges When Connecting to Public Internet

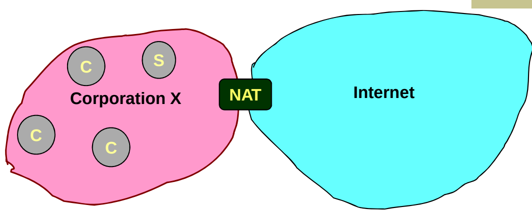


- Not enough IP addresses for every host in organization
  - Increasingly hard to get large address blocks
- Security
  - Don't want every machine in organization known to outside world
  - Want to control or monitor traffic in / out of organization

20

But not All Hosts are Equal!

C: Client
S: Server

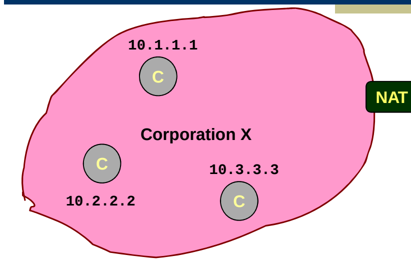


- Most machines within organization are used by individuals
  - For most applications, they act as clients
- Small number of machines act as servers for entire organization
  - E.g., mail server, web, ..
  - All traffic to outside passes through firewall

(Most) machines within organization do not need public IP addresses!

Reducing Address Use: Network Address Translation

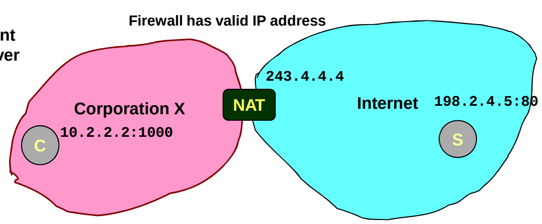
C: Client



- Within Organization: assign every host a private IP address
  - IP addresses 10/8 & 192.168/16 set aside for this
  - Route within organization by IP protocol, can do subnetting, …
- NAT translates between public and private IP addresses
  - Does not let any packets from internal nodes escape
  - Outside world does not need to know about internal addresses

NAT: Opening Client Connection

C: Client
S: Server

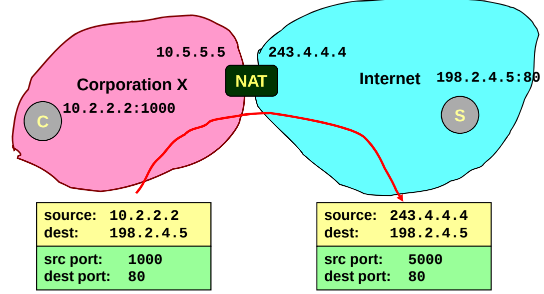


- Client 10.2.2.2 wants to connect to server 198.2.4.5:80
  - OS assigns ephemeral port (1000)
- Connection request intercepted by firewall
  - Maps client to port of firewall (5000)
  - Creates NAT table entry

Int Addr	Int Port	NAT Port
10.2.2.2	1000	5000

NAT: Client Request

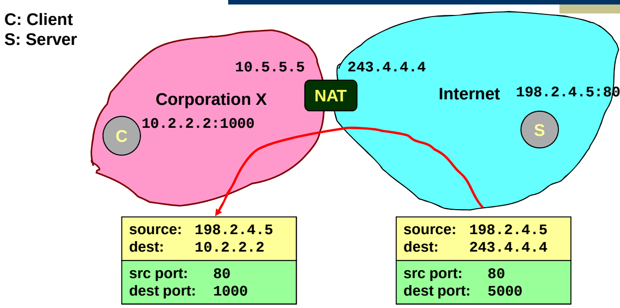
C: Client
S: Server



Int Addr	Int Port	NAT Port
10.2.2.2	1000	5000

- Firewall acts as proxy for client
  - Intercepts message from client and marks itself as sender

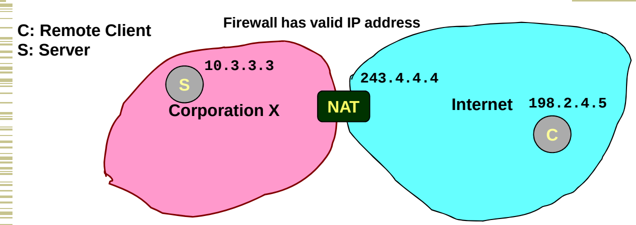
NAT: Server Response



Int Addr	Int Port	NAT Port
10.2.2.2	1000	5000

- Firewall acts as proxy for client
  - Acts as destination for server messages
  - Relabels destination to local addresses

NAT: Enabling Servers



- Use port mapping to make servers available

Int Addr	Int Port	NAT Port
10.3.3.3	80	80

- Manually configure NAT table to include entry for well-known port
- External users give address 243.4.4.4:80
- Requests forwarded to server

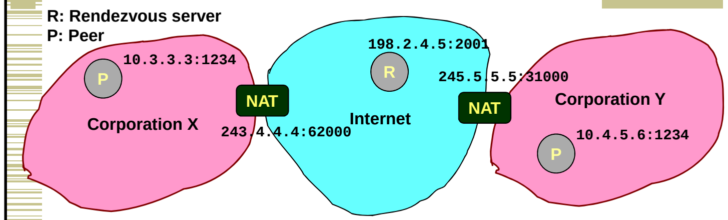
NAT Considerations

- NAT has to be consistent during a session.
  - Mapping (hard state) must be maintained during the session
    - Recall Goal 1 of Internet: Continue despite loss of networks or gateways
  - Recycle the mapping after the end of the session
    - May be hard to detect
- NAT only works for certain applications.
  - Some applications (e.g. ftp) pass IP information in payload - oops
  - Need application level gateways to do a matching translation
  - Peer-peer, multi-player games have problems – who is server?
- NATs are loved and hated
  - \- Breaks some applications
  - \- Inhibits deployment of new applications like (but so do firewalls!)
  - \+ Little NAT boxes make home networking simple
  - \+ Saves addresses, makes allocation simple

Often Combined with Firewalls

- NATs already help with security
  - Hides IP addresses used in internal network
    - Easy to change ISP: only NAT box needs to have IP address
    - Fewer registered IP addresses required
  - Basic protection against remote attack
    - Does not expose internal structure to outside world
    - Can control what packets come in and out of system
    - Can reliably determine whether packet from inside or outside
- But we have the disadvantages …
  - Contrary to the "open addressing" scheme envisioned for IP addressing
  - May be problematic for new application types, e.g., p2p
    - But network managers like it that way – "default off"

Many Options Exist for Peer-Peer



- NAT recognizes certain protocols and behaves as a application gateway
  - Used for standard protocols such as ftp
- Applications negotiate directly with NAT or firewall – need to be authorized
  - Multiple protocols dealing with different scenarios
- Punching holes in NAT: peers contact each other simultaneously using a known public (IP, port), e.g. used with rendezvous service
  - Use publicly accessible rendezvous service to exchange accessibility information
  - Assumes NATs do end-point independent mapping
- But remains painful!

Outline

- IP protocol
- NATs
- Tunnels

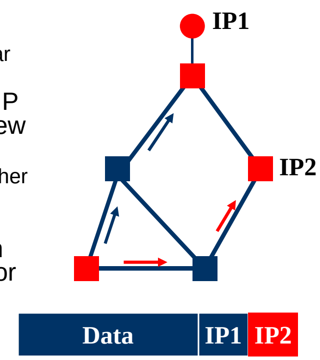
Motivation

There are many cases where not all routers have the same features or consistent state

- An experimental IP feature is only selectively deployed – how do we use this feature e-e?
  - E.g., IP multicast
- A few are using a protocol other than IPv4 – how can they communicate?
  - E.g., incremental deployment of IPv6
- I am traveling with a CMU laptop - how can I can I keep my CMU IP address?
  - E.g., must have CMU address to use services

Tunneling

- Force a packet to go to a specific point in the network.
  - Cannot rely on routers on regular path
- Achieved by adding an extra IP header to the packet with a new destination address.
  - Similar to putting a letter in another envelope
  - preferable to IP source routing
- Used increasingly to deal with special routing requirements or new features.
  - Mobile IP,..
  - Multicast, IPv6, research, ..

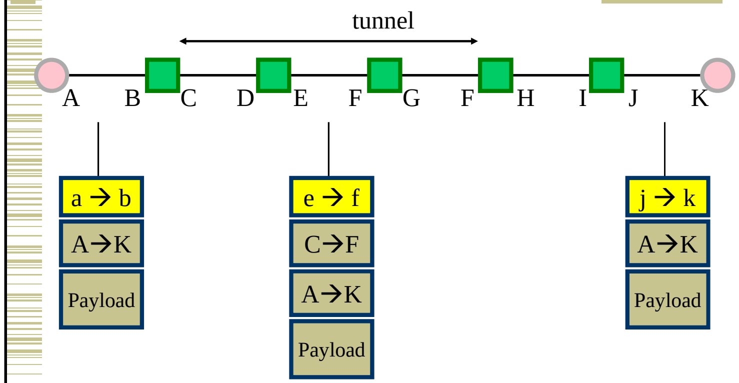


IP-in-IP Tunneling

- Described in RFC 1993.
- IP source and destination address identify tunnel endpoints.
- Protocol id = 4.
  - IP
- Several fields are copies of the inner-IP header.
  - TOS, some flags, ..
- Inner header is not modified, except for decrementing TTL.

V/HL	TOS	Length
ID		Flags/Offset
TTL	4	H. Checksum
Tunnel Entry IP		
Tunnel Exit IP		
V/HL	TOS	Length
ID		Flags/Offset
TTL	Prot.	H. Checksum
Source IP address		
Destination IP address		
Payload		

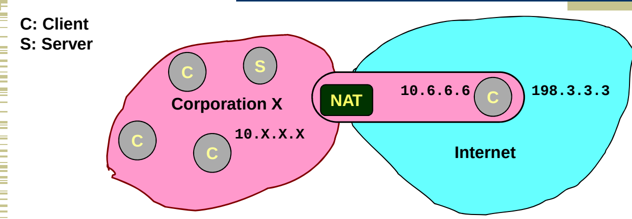
Tunneling Example



Tunneling Applications

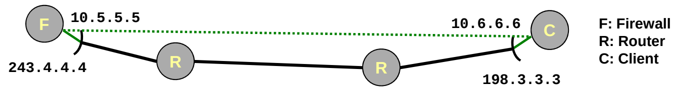
- Virtual private networks.
  - Connect subnets of a corporation using IP tunnels
  - Often combined with IP Sec (later)
- Support for new or unusual protocols.
  - Routers that support the protocols use tunnels to “bypass” routers that do not support it
  - E.g. multicast, IPv6 (!)
- Force packets to follow non-standard routes.
  - Routing is based on outer-header
  - E.g. mobile IP (later)

Extending Private Network

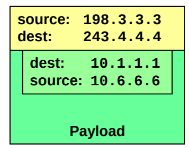


- Supporting Road Warrior
  - Employee working remotely with assigned IP address 198.3.3.3
  - Wants to appear to rest of corporation as if working internally
    - From address 10.6.6.6
    - Gives access to internal services (e.g., ability to send mail)
- Virtual Private Network (VPN)
  - Overlays private network on top of regular Internet

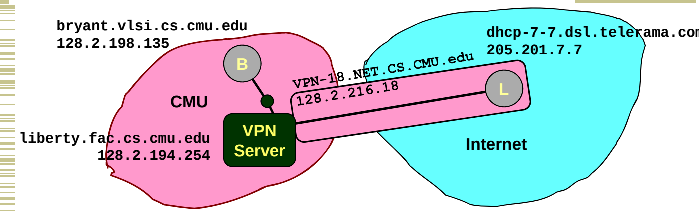
Supporting VPN by Tunneling



- Idea: client sets up tunnel to company's firewall
- Example: client wants to send packet to internal node 10.1.1.1
- Entering Tunnel
  - Add extra IP header directed to firewall (243.4.4.4)
  - Original header becomes part of payload
  - Possible to encrypt it
- Exiting Tunnel
  - Firewall receives packet
  - Strips off header
  - Sends through internal network to destination



CMU CS VPN Example



- CS has server to provide VPN services
- Operation
  - Running echo server on CMU machine 128.2.198.135
  - Run echo client on laptop connected through DSL from non-CMU ISP
- With VPN
  - server connected to 128.2.216.18 - VPN-18.NET.CS.CMU.EDU
- Without VPN
  - server connected to 205.201.7.7 - dhcp-7-7.dsl.telerama.com
- Effect
  - For CMU hosts, packets appear to originate from within CMU

Overlay Networks

- A network "on top of the network".
  - E.g., initial Internet deployment
    - Internet routers connected via phone lines
      - An overlay on the phone network
  - Tunnels between nodes on a current network
- Examples: IPv6 "6bone", multicast "Mbone".
- But not limited to IP-layer protocols…
  - Peer-to-peer networks, anonymising overlays
  - Application layer multicast
  - Improve routing, e.g. work around route failures

Important Concepts

- IP has a very simple service model
- IPv4 is a simple protocol, but there are issues
  - 32 bit address space is too small
  - Some messy features, e.g., fragmentation
  - Very simple "control" protocol
- NATs change to Internet addressing model
  - Have moved away from "everyone knows everybody" model of original Internet
- Firewalls + NAT hide internal networks
- VPN / tunneling build private networks on top of commodity network